Requested Patent: WO0180479A1

Title:

DELAYED COMMITMENT SCHEME TO PREVENT ATTACKS BASED ON COMPROMISED CERTIFICATES ;

Abstracted Patent: WO0180479 ;

Publication Date: 2001-10-25 ;

Inventor(s): WEN WU (US) ;

Applicant(s): WEN WU (US) ;

Application Number: WO2001US08654 20010413 ;

Priority Number(s): US20000197398P 20000414; US20000588215 20000606 ;

IPC Classification: H04L9/00 ;

Equivalents: AU5518301 ;

ABSTRACT:

A secure computer network (40) service based on public key cryptography enables communication between a client (28, 29 and 30) and a server (22) running on any node of the computer network to communicate critical information in a secret and integral manner against compromised certificate based attacks, such as an "ex-employee" attack on the Secure Socket Layer (SSL) handshake protocol. A two-step delayed commitment scheme is implemented, whereby in the first step, only partial information concerning a pre-master secret encrypted with server's public key is initially sent, and in a second step, full commitment to the pre-master secret encrypted with the server's public key is not performed until after an intervening communication from the server containing random information is received, thereby stopping the attacker from performing 1) switching of a compromised certificate and 2) re-encryption of a learned pre-master secret with the public key of the legitimate certificate during a single protocol run, thus preventing such an attack. The method integrates the two-step delayed commitment scheme to current authentication and key exchange protocols, requiring only minimum changes to the current protocol, yet providing enhanced security against attacks using compromised certificates.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 October 2001 (25.10.2001)

PCT

(10) International Publication Number
WO 01/80479 A1

(51) International Patent Classification⁷: H04L 9/00

(21) International Application Number: PCT/US01/08654

(22) International Filing Date: 13 April 2001 (13.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/197,398 14 April 2000 (14.04.2000) US
09/588,215 6 June 2000 (06.06.2000) US

(71) Applicant and
(72) Inventor: WEN, Wu [CN/US]; 806 Coleman Ave. #7, Menlo Park, CA 94025 (US).

(74) Agent: GUSS, Paul; 775 South 23rd Street, First Floor, Suite 2, Arlington, VA 22202 (US).

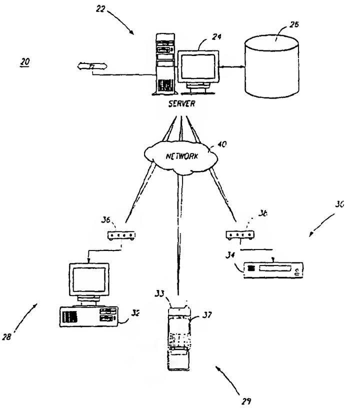
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DELAYED COMMITMENT SCHEME TO PREVENT ATTACKS BASED ON COMPROMISED CERTIFICATES



(57) Abstract: A secure computer network (40) service based on public key cryptography enables communication between a client (28, 29 and 30) and a server (22) running on any node of the computer network to communicate critical information in a secret and integral manner against compromised certificate based attacks, such as an "ex-employee" attack on the Secure Socket Layer (SSL) handshake protocol. A two-step delayed commitment scheme is implemented, whereby in the first step, only partial information concerning a pre-master secret encrypted with server's public key is initially sent, and in a second step, full commitment to the pre-master secret encrypted with the server's public key is not performed until after an intervening communication from the server containing random information is received, thereby stopping the attacker from performing 1) switching of a compromised certificate and 2) re-encryption of a learned pre-master secret with the public key of the legitimate certificate during a single protocol run, thus preventing such an attack. The method integrates the two-step delayed commitment scheme to current authentication and key exchange protocols, requiring only minimum changes to the current protocol, yet providing enhanced security against attacks using compromised certificates.

WO 01/80479 A1

DELAYED COMMITMENT SCHEME TO PREVENT ATTACKS BASED ON COMPROMISED CERTIFICATES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from U.S. Provisional Patent Application Serial. No. 60/197,398 filed on April 14, 2000 by Wu Wen, the disclosure of which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention:

The present invention relates to computer networks, and more particularly, to improving the security and privacy of communications between networked computers. The invention also has applicability to networked consumer appliances such as mobile phones, wireless PDAs and other networked home appliances.

Description of the Related Art:

To securely conduct business over the Internet, various cryptographic mechanisms and protocols have been proposed. Two very important objectives of security protocols are that of authentication and key exchange, in which the communicating parties establish the identity of their respective counterparts and exchange a secret key. The secret key is used to conduct future private communications between the authenticated parties.

Protocols designed to achieve the above two objectives usually employ a mixture of public key and symmetric key cryptography for scalability and efficiency. A simple authentication and key exchange protocol using password and public key devices is disclosed by Halevi and Krawczyk, "Public-key Cryptography and Password Protocol," ACM Transactions on Information and System Security, vol. 2, no. 3, pp. 230-268 (1999), the disclosure of which is incorporated herein by reference. A more sophisticated authentication and key exchange protocol is the Secure Socket Layer (SSL) protocol, disclosed in United States Patent No. 5,657,390 to Elgamal et al., the full disclosure of which is expressly incorporated into the present application by reference. A successor protocol to SSL is the Transport Layer Security (TLS) protocol, details of which have been disclosed by T. Dierks and C. Allen, "The TLS protocol: Version 1.0," RFC 2246, January 1999, the disclosure of which is also incorporated herein by reference.

The advantage of public key cryptography over symmetric key cryptography for authentication and key exchange is that there is no need for a shared secret between the communicating parties to exist prior to the initiation of the protocol. In fact, the very purpose of the authentication and key exchange protocol is to establish such a shared secret securely without any prior communications between each of the parties. Commonly, a randomly chosen key (pre-master secret) N_C^* is generated by an initiator C and sent securely to the receiver S after the pre-master secret is encrypted with the receiver's public key K_S^+. Only a receiver in possession of the corresponding private key K_S^- can obtain the key N_C^* thus achieving authentication and key exchange. The pre-master secret may then be used by both C and S for generating a symmetric key (master secret) K_{cs} under which subsequent communications are conducted.

Unfortunately, things are rarely so simple. The security of the authentication and key exchange protocol now rests on the security of the public-private key pair. This raises two significant questions: (1) Does the public key K_S^+ actually belong to the named party S who owns the private key K_S^-? (2) Is the private key K_S^- corresponding to the public key K_S^+ still secure? If someone other than S also has access to the private key K_S^-, he can decrypt any messages encrypted with K_S^+ and hence the security of the protocol is broken.

To remedy the first problem, and to ensure that the public key K_S^+ and the name of its owner S are securely binded, the concept of certificates was introduced, as described by L. Kohnfelder, "Towards a Practical Public-key Cryptosystem," Technical Report, MIT (1978). With this cryptosystem, a certificate $\{S, K_S^+\}_{K_{CA}^-}$ is provided comprising the pair name S and the public key K_S^+ signed by a trusted third party CA known as the Certificate Authority using *CA*'s private key K^{-}_{CA}.

The solution to the second problem, which is known as freshness proof, has conventionally required additional support provided by the Public Key Infrastructure (PKI). If the private key corresponding to the public key contained in the certificate becomes known to a non-trusted third party, the certificate is said to have been compromised. To prove that a given certificate is not compromised, a Certificate Revocation List (CRL) needs to be consulted before the certificate can be trusted and used.

Thus, to prove the validity of the certificate, the *CA*'s verification key can be used, and since validity is a static property, validity checking needs only to be performed once. By contrast, freshness proof is a dynamic property and must be checked every time the certificate is used.

Many practical and social problems face today's PKI. For example, there is no efficient and practical way to cross-certify certificates issued by different countries, governments or even companies. Further, users are not familiar with the intricate problems compromised certificates can engender. Thus, an owner of a certificate is not only responsible for maintaining absolute secrecy of his own certificate, but he is also responsible for constantly checking the certificates of the persons he communicates with. The SSL/TLS protocol is so far the most robust authentication and key exchange protocol designed, but clearly, more ingenious protocols are needed to deal with the problems discussed above.

FIG. 1 is a diagrammatic illustration of a client-server system 20 to which the teachings of the present invention are applicable. More specifically, the system is constituted by a headend facility 22 with a server terminal 24 and associated file storage 26. The file storage can include any of various interactive or non-interactive content, such as HTML documents, image and/or sound files, CGI scripts, etc. which are delivered by the server to various multiple participants 28, 29 and 30 which request the services provided by the headend facility 22. The headend facility 22 is typically interconnected over a high-speed T1 line with other similar servers, so that the headend facility 22 serves not only as a content provider but also as the gateway through which access to any of multiple servers can be established.

The participants 28, 29 and 30 are each equipped with corresponding computing units, wherein such units may be embodied as different types of devices. For example, the participant computing units are illustrated as a desktop computer 32, a mobile telephone unit 33, and a home appliance device 34 one example of which is a set-top box having a digital video recording capability. It shall be understood that other digital processing mechanisms that are capable of handling digital data supplied by the server may also be used. Similarly, the devices are expected to upload data to the server 24 which may include sensitive and/or confidential information requiring a secure connection.

The participant units 32, 33 and 34 are interconnected to the server 24 via a network, represented by network cloud 40. The network 40 may be in the form of a wireless network, such as a satellite or cellular phone network, a wire-based network, such as low-bandwidth telephone lines, or a higher-bandwidth cable or DSL network, or any combination thereof.

Typically, the desktop computer 32 and home appliance 34 will connect to the network 40 over modem devices 36, 38 which may be conventional or high-speed cable modems, or any other known connecting devices, such as wireless or satellite systems, for establishing connections to the network 40. Of particular interest to the present invention are mobile telephones 33 which establish a wireless connection with the network 40 and operate using so-called WAP and I-mode standards in which Internet functions are made possible, such as sending and receiving of email messages, or viewing and interacting with Internet content via an integrated browser program and display screen 37 incorporated into the mobile telephone unit 33.

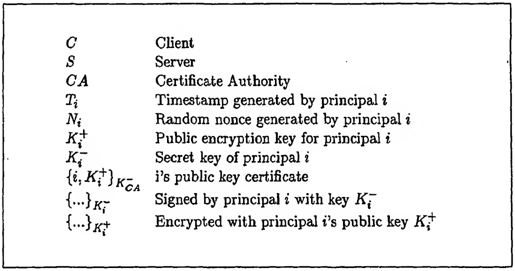
Although not illustrated, small handheld computing devices known as Portable Digital Assistants (PDAs) may also be used.

Such devices also possess Internet functions such as email and web browsing and establish connections to the network 40 using wireless systems or modem lines.

For certain types of content which are delivered by or via the server 24, for example when connecting to an on-line bank, it is necessary for a secure connection with a server to be established. Current SSL/TLS technology provides a program layer for managing the security of message transmissions over the network 40. As described in U.S. Patent No. 5,657,390, the programming for keeping messages confidential is contained in a program layer between an application (such as a web browser) and the Internet's TCP/IP layers. SSL/TLS is an integral part of the known Netscape and Internet Explorer browsers., so that when a web site on a given server is identified as requiring a secure connection, SSL/TLS can be enabled.

FIG. 2 shows the steps involved in the handshake protocol implemented under the conventional SSL/TLS method. The basic and most widely implemented mode of the SSL/TLS handshake protocol is the "named-server anonymous-client" version of the protocol, in which only the server, such as an Internet shopping mall, is authenticated and uses a certificate. The basic function of the protocol is to generate and transfer a shared "pre-master secret" between an anonymous client and a named server. As described in U.S. Patent No. 5,657,390, the "pre-master secret" (referred to under different terminology in Elgamal et al. as a "master key") is typically a randomly generated number which is used by the client and server to produce a symmetrical key (the so-called "master secret") which later is employed to actually encrypt/decrypt all data to be transferred through a known type of IP sockets connection.

The following notations are used in describing the handshake protocol shown in FIG. 2:



C	Client
S	Server
CA	Certificate Authority
T_i	Timestamp generated by principal i
N_i	Random nonce generated by principal i
K_i^+	Public encryption key for principal i
K_i^-	Secret key of principal i
$\{i, K_i^+\}_{K_{CA}^-}$	i's public key certificate
$\{...\}_{K_i^-}$	Signed by principal i with key K_i^-
$\{...\}_{K_i^+}$	Encrypted with principal i's public key K_i^+

Table 1: Notations Used to Describe SSL/TLS Handshake Protocol

More specifically, as described in more detail by T. Dierks and C. Allen, "The tls protocol: Version 1.0" referred to above, the basic protocol is made up of six messages transferred between client and server. In message M_1, the client C terminal sends a timestamp T_c and a client-side nonce N_c to the server S. A "nonce" is a known term in the art meaning a random number used to ensure channel integrity. In reply to message M_1, the server returns a message M_2 similarly made up of another timestamp T_S and a server-side nonce N_S. Directly thereafter, in message M_3, the server sends its authenticated certificate $\{S, K_S^+\}_{K_{CA}}$ to the client which, as noted above, is made up of the server's public key binded to the server name S and authenticated by the digital signature of a trusted Certificate Authority.

Message M_4 is a critical step in the handshaking process, since it is at this stage that the "pre-master secret" N_C^* generated on the client side is transmitted to the server S, with the pre-master secret being encrypted under the server's public key K_S^+. Since the pre-master secret N_C^* is used for generating the actual symmetrical key ("master secret") K_{CS} under which all subsequent sensitive communications flow between the client and server, one should at once suspect that if the pre-master secret were to fall into the hands of an imposter, the integrity of the handshake could be compromised. Precisely how such a compromise can occur, which is essential to an understanding of the present invention, shall be described in much greater detail to follow.

In particular, the master secret K_{CS} is calculated as a function of the pre-master secret N_C^*, and the client and server nonces N_C and N_S, according to the following equation:

$$K_{CS} = f(N_C, N_S, N_C^*) \qquad \ldots (1)$$

Messages M_5 and M_6 are confirmatory steps performed under the encryption of the master secret K_{CS} (generated from the pre-master secret N_C^* as described above) to verify the integrity of the communication between the client and server. More specifically, in its server_done message M_5, the server sends a hash function $H(K_{CS}, CS_5, [M_1, M_2, M_3, M_4])$ of the master secret K_{CS}, a tag CS_5 indicating the server_done protocol stage, and all preceding messages M_1, M_2, M_3 and M_4 sent between the server and the client, wherein the hash H is encrypted under the master secret K_{CS}. In a complementary client_done message M_6, the client sends a hash function $H(K_{CS}, CS_6, [M_1, M_2, M_3, M_4])$ of the master secret K_{CS}, a tag CS_6 indicating the client_done protocol stage, and all preceding messages M_1, M_2, M_3 and M_4 sent between the client and the server, wherein again the hash H is encrypted under the master secret K_{CS}. The purpose of messages M_5 and M_6 is to prevent attacks that are based on interception and faking of messages, by enabling the respective parties to check the hash of all previous messages against messages known to have been already sent and received.

Having confirmed the integrity of the above messages, the client and server are now set up to encrypt all communications from this point on using the master secret K_{cs} (sometimes also referred to as the "session key" or "master key") which is a symmetrical key used between the client and server.

The SSL/TLS handshake protocol has as its underpinning the reliability of the server's public key, as certified by the certificate $\{S, K_S^+\}_{K_{ca}}$. Unfortunately, certificates alone are not enough to provide the required security for electronic commerce in an open environment such as the Internet. A set of protocols along with methods to guarantee that certificates are fresh are required. Hence, the concept of a Public Key Infrastructure (PKI) was envisioned so that the validity of all certificates could be verified by the possession of the root verification key. Further, to solve the problem that valid certificates can become invalid over time due to various reasons, a Certificate Revocation List (CRL) was also provided in the PKI. In theory, any principal who believes that his certificate has been compromised can revoke that certificate by putting it on the CRL. An entity who wishes to make a secure communication with the principal can check the validity of its certificate by searching through the CRL. However, as a practical matter, having to comb through all the CRLs before knowing if a given certificate is valid is an impossible burden. Moreover, it is foreseeable that a CRL in the possession of a communicating entity (for example at an Internet cafe) may not be the most recently available, and hence is unreliable.

Although attempts to alleviate the problem of requiring the client to comb through all the CRLs have been proposed, such as the Online Certificate Status Protocol (OCSP), as of today, there is still no solution which can provide an absolute guarantee for certificate freshness proof at any time. On the other hand, commercial products such as the popular Netscape browser or the Internet Explorer have incorporated certificate based authentication protocols such as the above-described SSL and TLS. However, in such basic applications of SSL/TLS, freshness of the certificates is not checked against a Certificate Revocation List, or if checking against a CRL is done (such as in the Internet cafe example) it may be conducted using a local CRL which has become outdated, thereby exposing such systems to potential security holes. Moreover, even if access to fresh CRLs could be provided, it would still be cumbersome and expensive to link to such CRLs and provide proof of freshness for all certificates, particularly for small devices such as mobile phones and PDAs.

As described above, public key based security protocols depend on the freshness of the server certificate for their reliability. Since the SSL/TLS protocol does not itself check the freshness of the certificates used under the protocol, the conventional wisdom is that freshness should be guaranteed by some other protocol, such as OCSP. However, in reality, such guarantees can be very difficult and expensive to implement. With even wider applications of public key certificates, such as proposed by the public key infrastructure (PKI), it is clear that the number of certificates will increase, the areas where such certificates are expected to be used will broaden, automated uses of certificates will become more standardized, and as a consequence, CRLs will become even more unmanageable. Such facts lead to the conclusion that as use of certificates increases, the reality of compromised certificates will become more widespread and more difficult to discover and manage.

In most of the discussions to date on analysis of security protocols, the case of compromised certificates has been either overlooked or the present state of affairs declared unsafe. The security of public-key based protocols such as SSL/TLS, which have been analyzed under assumed conditions of perfectly fresh certificates only, must be reevaluated in light of the existence of compromised certificates.

A serious security lapse has been discovered by the inventor in the "named-server anonymous-client" version of the SSL/TLS protocol most widely used in internet commerce applications. Particularly, the conventional SSL/TLS protocol is vulnerable to an attack by an individual (for example, an "ex-employee") who comes into possession of a compromised server certificate which might not yet have been placed on the

Certificate Revocation List (CRL) to which the client application refers, even though the server uses a new certificate. A compromised server certificate also enables an attack against various client applications which, for reasons of economy or speed, simply do not refer to any CRL. Such an attack allows the illegitimate certificate holder to eavesdrop on subsequent communications between the client and the server because, as shall be described later, the intruder can intercept the pre-master secret and thereby derive the master secret used to decrypt all communications between the client and server.

SUMMARY OF THE INVENTION

It is therefore a principal object of the invention to provide an improved public key security protocol which provides extra protection against the problem of compromised certificates.

The present invention aims to resolve the problem of freshness of certificates by requiring the client and server, during the handshaking process, to undertake additional steps which will reveal if a compromised certificate is being used. This offers a significant advantage in that the client is not required to refer to any CRL to determine certificate freshness.

The attack on the conventional SSL/TLS protocol requires an attacker to do two things: a) first, switch the valid certificate with a compromised certificate and learn the pre-master secret from the client; and b) once learned, then re-encrypt the pre-master secret with the server's legitimate public key so the server is unaware that the switch has taken place.

In accordance with the present invention, additional steps to the handshaking protocol provide for a two-step delayed commitment which effectively prevents the attacker from accomplishing these two things as the same time. The attacker can only do one but not the other, and thus the attack is stopped.

The above and other objects, features and advantages of the present invention will become apparent from the following description when taken in conjunction with the accompanying drawings in which preferred embodiments of the present invention are shown by way of illustrative example.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a basic network configuration in which different types of client devices can establish secure connections with a server terminal over a network.

FIG. 2 shows the steps involved in the handshake protocol implemented under the conventional SSL/TLS method.

FIG. 3 is a description of the steps making up the conventional SSL/TLS protocol, together with detailing the steps by which an intruder in possession of a compromised certificate can learn the master secret K_{CS}.

FIG. 4 illustrates a basic first-step approach toward preventing the "ex-employee" attack against the named-server anonymous-client version of the SSL/TLS protocol.

FIG. 5 shows a further development of the approach illustrated in FIG. 4, and illustrates the steps of a handshake protocol possessing greater integrity which overcomes a potential attack on the embodiment shown in FIG. 4.

FIG. 6 shows a still further embodiment, based on the same principles as the handshake protocol depicted in FIG. 5, but modified so as to minimize the changes necessary to integrate the present invention with the current conventional SSL/TLS protocol.

FIG. 7 shows yet another embodiment of an improved handshake protocol according to a different method from that shown in FIGS. 5 and 6.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Before entering into a detailed discussion of the improvements to the handshaking protocol implemented by the present invention, it is important first to discuss some basic assumptions about certificate verification and compromised certificates, and thereafter to describe the types of attacks possible under known SSL/TLS schemes, which attacks are defeated using the teachings of present invention.

As described in the background section, two aspects of certificate security need to be considered. The first is certificate validity, which is a static property and can be verified using the CA's verification key, and need only be checked once when the certificate is first used. This prevents any attempt to substitute an attacker's public key for that of the legitimate owner, because the attacker cannot generate the signature necessary to bind his public key to the legitimate server name. The second aspect of public key security is the freshness property. A legitimate certificate becomes compromised when the private key corresponding to the public key contained in the certificate is known to any non-trusted third party. The freshness property, moreover, is dynamic and must be checked every time the certificate is used. Theoretically, a certificate revocation list (CRL) can be provided which publishes all revoked certificates. However, for reasons mentioned above, reliable access to the CRL cannot be assured. In reality, therefore, it is both technically difficult and financially very expensive to provide freshness proof.

For example, a client browser used in an Internet cafe may not be set up to check the freshness of the server certificate as required. Further, in browsers used in cellular telephones, or in portable digital assistants (PDAs) which have internet-access functionality, because of program size and data handling restrictions in such environments, it is not easily feasible to provide a means of access to a CRL, and hence in such applications, freshness of certificates is generally left unchecked.

On the other hand, when a server's certificate $\{S, K_S^+\}_{K_{CA}^-}$ becomes compromised, this means that the private key (i.e. the key which enables a person to decrypt messages encrypted under the public key of the server K_S^+) is known to someone other than the owner of the certificate. Thus, although it is not possible for an ill-willed third party to fake a certificate with a proper *CA* signature, it is nevertheless possible for the third party to acquire the ability to decrypt messages encrypted under the certified public key K_S^+. Moreover, while it may be assumed that the owner of the certificate knows when his certificate has become compromised and immediately acts to revoke the certificate by placing it on the CRL, the practical reality is that not all clients will have access to the most recent CRL or may not have the resources to check the certificate against any CRL at all.

The fact that today's Internet is shared and open leads to various opportunities for determined attackers bent on gaining unauthorized access. Such opportunities include abilities to: a) overhear and store a copy of the messages transmitted between client and server; b) intercept or steal the message from its intended receiver; c) decrypt a message encrypted with an encryption key for which the intruder has its corresponding decryption key; d) store such intercepted messages and replay them at any time; and e) make up new messages using a learned secret such as a stolen nonce.

There are three types of attacks which can be considered: 1) An impersonation attack in which the intruder attempts to conduct a protocol run with the intruder interposing himself in the communications between the parties; 2) the passive eavesdrop in which the intruder comes into possession of the master secret used in the communication channel, so that he does not need to actively manipulate the channel in order to gain access to the messages; and 3) a so-called MIM (man-in-the-middle) attack in which the intruder must actively manipulate the communication channel to gain access to messages; for example, he must use different keys to talk to each party throughout the communication while the two parties assume one key is being used.

It will be appreciated that an MIM attack is very difficult to maintain for any sustainable time period. The passive eavesdrop is much easier to carry out than a sustained man-in-the-middle attack, since the latter requires the attacker to decrypt and re-encrypt all communications between the client and server, which is an almost impossible feat. However, if a short impersonation attack involving interception and faking of messages can be conducted across a few message transfers, which enables the intruder to come into possession of the master secret

K_{cs}, thereafter the intruder will easily be able to siphon and store all subsequent communications between the client and server and decrypt them at his leisure. Such a scenario raises even greater concerns when one considers that in the typical electronic commerce transaction, critical sensitive data, such as the user's credit card number and expiration date, are transmitted under encryption by the master secret K_{cs} after the handshaking protocol has taken place.

Moreover, if an intruder possesses the master secret K_{cs} for the record layer, he will be able to insert, delete or modify messages at will because the receiver, whether it be the client or the server, will decrypt and verify such altered messages while believing them to be authentic. This type of attack is particularly dangerous for financial transactions. In addition to destroying confidentiality, the ability to alter messages which are supposed to be secure destroys at its core the integrity and non-repudiation properties of the SSL/TLS protocol.

Next, the attack on the named-server anonymous-client SSL/TLS protocol, which is the standard used in electronic commerce transactions between a client browser and a server such as an Internet shopping mall, shall be described. More specifically, FIG. 3 details the steps showing how an intruder with a compromised certificate can learn the master secret K_{cs} even if the server has updated its own certificate.

In FIG. 3, on the left-hand side of the figure, the direction of the communication between client C and server S is shown, $C \rightarrow S$ indicating a communication from the client C to the server S and $S \rightarrow C$ indicating a communication from the server S to the client C. Further, a message in the form of $\ldots \rightarrow (X)I$ indicates that a message intended for X is intercepted by I, whereas a message in the form of $(X)I \rightarrow \ldots$ indicates a message faked by I and misrepresented as originating from X.

The prime marker ′ in $\{S,K'^{+}_{S}\}_{K_{CA}}$ indicates a compromised certificate of S, while a fresh and valid certificate of S is indicated by $\{S,K^{+}_{S}\}_{K_{CA}}$. M'_{i} is a message intercepted by the intruder and M''_{i} is a message which is faked by the intruder. Such messages are otherwise identical in format with M_{i}.

As discussed above, one scenario in which an intruder may come into possession of a compromised certificate is that of a disgruntled or former employee of the service provider, and thus is a sophisticated individual who has access to, or is able to acquire through malfeasance, the server's former private key K'^{-}_{S}. Since the client is unable to reliably confirm the freshness of the certificates, the intruder deceives the client intc decrypting its messages under the server's former, now compromised, public key K'^{+}_{S} which the intruder can readily decrypt using K'^{-}_{S}.

Thus, the intruder is able to effect the attack as follows:

Messages M_1 and M_2 are sent in plain text in identical fashion to the prior art SSL/TLS protocol discussed previously. In message M'_3 the valid certificate for S is intercepted, and next, in message M''_3, the intruder substitutes the valid certificate $\{S,K_S^+\}_{K_{CA}}$ with the compromised certificate $\{S,K'^+_S\}_{K_{CA}}$. Hence, in the following message M'_4, the client C unwittingly encrypts the pre-master secret N_C^* under the compromised public key K'^+_S enabling the intruder, upon interception of the message M'_4, to decrypt and learn the pre-master secret. The intruder then re-encrypts the pre-master secret under the server's valid public key K_S^+, and sends the pre-master secret N_C^* on to the server S in the faked message M''_4. At this point, neither the client nor the server is aware of what has taken place, and more importantly, since the intruder is now in possession of N_C, T_C, N_S, T_S and N_C^*, the intruder is able to calculate the master secret K_{CS}.

Interception and faking of the last four messages M'_5, M''_5, M'_6 and M''_6 is now possible, and is in fact required, since both C and S are expected to maintain a consistent record of past messages. Note that in the above-described attack, from either one of the client's or the server's sole perspective, the past messages appear to be consistent, despite the different versions of M_3 and M_4 kept by C and S respectively.

The improved handshake protocol which serves to defeat the above type of attack shall next be explained in detail. The above-described "ex-employee" attack against the named-server anonymous-client version of the SSL/TLS handshake protocol succeeds because the intruder learns the pre-master secret N_C^*, along with the other two nonces N_C and N_S which are transmitted in plain text. From this information, the intruder can calculate the master secret K_{CS} and succeed in faking the client_done and server_done messages. Because of the three facts that (1) the client is anonymous and therefore cannot generate authenticated messages, (2) the server certificate is itself compromised, and (3) adequate checking of certificate freshness against a CRL is infeasible, it may appear impossible to prevent this type of attack without requiring the client to have its own certificate.

FIG. 4 illustrates a basic first-step approach toward preventing the "ex-employee" attack against the named-server anonymous-client version of the SSL/TLS protocol. The notations used in FIG. 4 are the same as shown in FIGS. 2 and 3 as well as Table 1.

As shown in FIG. 4, to provide a certain unique identity for the "anonymous" client, a new private-public key pair is generated by the client for each session. The public key K_C^+ of the key pair, which is sent in the client_hello message M_1 together with the client nonce N_C and timestamp T_C, will be used by the server to encrypt the server nonce N_S in the server_hello message M_2.

From equation (1), the master secret K_{CS} is derived from N_C,

N_s and N_c^*. However, according to the protocol shown in FIG. 4, even if the intruder learns N_c^* because he has control over the compromised server certificate, the intruder is still unable to learn the server nonce N_s encrypted with the client public key K_c^+ because he does not have the private key generated by the client for this particular session.

However, the intruder can nevertheless interpose himself between the client-server communication in the initial phase of the handshake while generating his own public key pair, to replace the client public key with that of his own, as follows:

$$\begin{array}{lll} C \rightarrow I(S): & (N_C, T_C, K_C^+) & (M_1) \\ I(C) \rightarrow S: & (N_C, T_C, K_C'^+) & (M_1') \\ S \rightarrow I(C): & (\{N_S\}_{K_C'^+}, T_S) & (M_2) \\ I(S) \rightarrow C: & (\{N_S\}_{K_C^+}, T_S) & (M_2') \end{array}$$

Such an attack enables the intruder to learn the server nonce N_s and then carry on with the rest of the "ex-employee" attack described in FIG. 3, ultimately leading to knowledge of the master secret K_{cs}, from whence the passive attack can be implemented. Thus the basic method of FIG. 4, while erecting certain additional barriers that the intruder must overcome before learning the master secret, should not be considered a fail-proof solution.

A handshake protocol possessing greater integrity, and which foils the above attack, is shown in FIG. 5. In the protocol shown in FIG. 5, the server_hello message is broken into two parts, messages M_2 and $M_{4.1}$. The first part consists of a hash of the encrypted server nonce $H(\{N_S\}_{K_C^+})$ whereas the second part of the message, bearing the encrypted server nonce $\{N_S\}_{K_C^+}$ itself, is sent after message M_4.

An intruder seeking to raise an attack upon the protocol shown in FIG. 5 can, of course, still replace the client's public key with that of his own in message M_1, however, the intruder is unable to learn the server nonce N_S at step M_2 and re-encrypt it with the client's public key because he only possesses the hash of the encrypted server nonce. Before the client will agree to send message M_4, the intruder must make one of two choices: (1) either pass the hash of the encrypted server nonce $H(\{N_S\}_{K_C^+})$ to the client as is, or (2) invent a completely new server nonce N'_S and send a hash of the encrypted faked nonce to the client. In the case of (1), the intruder must also pass $M_{4.1}$ as is, in which case the client wil discover that M_2 and $M_{4.1}$ are encrypted with a different public key than the one he has. In the case of (2), the intruder is forced to invent both another private-public key pair and a faked server nonce, since the client expects both the hash of the encrypted server nonce $H(\{N_S\}_{K_C^+})$ and the encrypted server nonce $\{N_S\}_{K_C^+}$ itself before committing to the secure channel. As a result, the client and server will be using different server nonces N'_S and N_S respectively. Moreover, from equation (1), we see that the master secret K_{CS} calculated by the server and K'_{cs} calculated by the client will be different.

Thus, the intruder is forced into sustaining a man-in-the-middle situation, which is extremely difficult to accomplish. Otherwise, the protocol will stop because the messages will not make sense after being decrypted. The intruder is therefore required to intercept each message from the server, decrypt them with K_{cs}, and then re-encrypt them with K'_{cs}. The intruder must also intercept each message from the client, decrypt them with K'_{cs}, and then re-encrypt them with K_{cs} . As described previously, this kind of attack is far more difficult than the passive eavesdrop attack, in which the intruder need only decrypt subsequent communications after learning the master secret K_{cs} and in which the intruder merely has to record the encrypted communications and then decrypt them off-line at a later time. Accordingly, the improved handshake protocol offers extra security by rendering the passive eavesdrop attack unattainable.

FIG. 6 shows a still further embodiment based on the same principles as the handshake protocol depicted in FIG. 5. More specifically, tc minimize the changes that need to be made to the current conventional SSL/TLS handshake protocol, the approach described in relation to FIG. 5 is integrated into the conventional protocol by adopting the following tactics.

First, we let K_c^+ be N_c . Instead of generating a random nonce, the client is required to generate its own private-public key pair and send the public key as the nonce N_C. In other words, the nonce N_C itself doubles in function as the client's public key K_C^+.

Secondly, we let $H(\{N_S\}_{K_C^+})$ be N_S. In addition to generating a random nonce N_S, the server is required to encrypt the nonce N_S using the client's public key K_C^+, received in message M_1, and send it as N_S. Thus, the nonce N_S serves the dual function of enabling the sending of a hash of an encrypted server-nonce to the client.

Notice that in the integrated version of the handshake protocol, as shown in FIG. 6, N_C and N_S are generated based on the above-described tactics, wherein N_S is an original nonce encrypted and sent in the form of the server nonce. Moreover, N_S is employed, along with the client nonce N_C and the pre-master secret N_C^*, for generating the master secret, so that the session master secret K_{CS} is still produced as a function of N_C, N_S and N_C^* as in equation (1):

$$K_{CS} = f(N_C,\ N_S,\ N_C^*) \qquad \ldots (1)$$

Notice also that the client learns the actual value of N_S only after message $M_{4.1}$ instead of learning it after message M_2 as in the conventional protocol. Thus, commitment to the protocol, and generation of the master secret K_{CS}, is delayed until both steps of a two-step commitment scheme have been implemented.

FIG. 7 shows yet another embodiment of an improved handshake protocol, according to a somewhat different method from that shown in FIGS. 5 and 6. More specifically, the method shown in FIG. 7 does not require the client to generate a private-public key pair as in the preceding embodiments.

To impose a computational burden on the client device to generate its own private-public key pair may be too expensive for small devices such as mobile phones or portable digital assistants (PDAs). It also has the disadvantage of forcing the client to select a cipher suite before one can be agreed to with the server after exchange of the second message, whereas it is a feature of conventional SSL/TLS that the particular encryption scheme used should generally be selected by the server based on the client's available capabilities and needs.

In the method shown in FIG. 7, the intention is to protect the client-generated pre-master secret N_C^*. This goal is achieved by adding a new server-generated nonce N_S^* to the protocol, thereby forcing the intruder to fake a different N_C^*, so that the master secret N_{CS} calculated therefrom will be different. Therefore, the method shown by FIG. 7 is similar to that of FIGS. 5 and 6 in that the aims of both methods are the same. However, the method shown by FIG. 7 can be implemented without requiring the client to perform the burdensome operation of generating its own public-private key pair, as well as avoiding the disadvantage of restricting beforehand choices for the cipher suite before negotiating with the server.

Note that, according to the method of FIG. 7, the session "master secret" is now produced as a function of N_C, N_S, N_S^* and N_C^* as follows:

$$K_{CS} = f(N_C, N_S, N_S^*, N_C^*) \qquad \ldots (2)$$

In message $M_{4.0}$, a hash of the client-generated pre-master secret N_C^*, which is encrypted under the server's secure public key, is sent to the server. Further, the server expects message $M_{4.0}$ before it will send out the new server-generated nonce N_S^* in step $M_{4.1}$. Thus, even in the event that an intruder intervenes and intercepts message $M_{4.0}$, at this point, since the intruder cannot determine N_C^*, the intruder must choose between (1) passing the hash of the encrypted pre-master secret as is, or (2) fake a different N_C^*. If the intruder tries (1), the server will discover that the pre-master secret N_C^* was not encrypted with the server's current public key after message

M_4. On the other hand, if the intruder attempts (2), there will be two versions of N_C^* (one real and one faked) known to the client and server, respectively, and thus we are back to the argument that the intruder is forced to commit to an active and sustained man-in-the-middle attack rather than the passive eavesdrop.

It shall be understood that the embodiment shown in FIG. 7 possesses the advantage that the client_hello and server_hello messages M_1 and M_2 are not changed from the conventional SLL/TLS protocol, so that negotiation with the server over a cipher suite, based on client capabilities and needs, can be made. There is also no need for the client to expend the computational burden of generating its own private-public key pair.

The method shown in FIG. 7, however, does require the addition of two new messages $M_{4.0}$ and $M_{4.1}$ to the current protocol, as opposed to only a single new message as in the embodiment shown in FIGS. 5 and 6. Further, as shown in Equation (2) above, the function for calculating K_{CS} differs from the conventional SSL/TLS protocol. Despite these differences from the current SSL/TLS protocol, implementation of the handshake protocol shown in FIG. 7 can be easily facilitated, since use of the new messages $M_{4.0}$ and $M_{4.1}$ can be offered and declared as an option during exchange of the client_hello and server_hello messages (i.e. during cipher negotiation), so that the client and server can optionally choose to engage in the improved handshake protocol or simply ignore it. Such an option makes the improved protocol backwards compatible with current systems which may not be prepared to operate using the improved handshake.

A similar type of optional election of the protocol can also be used with the embodiments of FIGS. 5 and 6. For example, should the client and server elect not to use the improved protocol, the client's public key K_C^+ in message M_1 is simply ignored, and subsequent messages are transferred in accordance with the conventional SSL/TLS protocol, thus achieving the same backwards compatibility with existing systems.

The embodiments of the invention shown in FIGS. 5 through 7 all effectively provide for a "two-step delayed commitment" scheme. More specifically, as described in connection with FIG. 3 above, it is possible for an attacker to intercept and switch the fresh certificate sent in message M_3, and use the private key of the compromised certificate to learn the pre-master secret N_C^* after intercepting M_4. But to make this attack work, the attacker must also re-encrypt the stolen pre-master secret N_C^* with a legitimate public key. The two-step delayed commitment is designed to prevent an attacker from doing both at the same time.

In other words, referring again to FIG. 7, even if the attacker decides to switch the certificate in step M_3, the attacker is unable to generate a hash of the legitimate pre-master secret encrypted with the legitimate public key of the server. While it is still possible for the attacker to generate a hash of a faked pre-master secret encrypted with the legitimate public key of the server, this will lead to different master secrets calculated by the client and server respectively. Of course, if the attacker doesn't switch the certificate in step M_3, he is incapable of learning anything.

The "commitment" in the "two-step delayed commitment" scheme refers to the actions taken for committing to a certain pre-master secret N_C^* by encrypting it using the server's public key. According to the present invention, such a commitment is divided into two discrete steps. In the first step, only partial information is transmitted. For example, in the embodiment of FIG. 7, a hash of the encrypted pre-master secret is sent in message $M_{4.0}$. Only after receiving the second server nonce N_S^* which serves to effectively separate these two steps, a full committed message with all information, (i.e. the encrypted pre-master secret of N_C^*) is then transmitted in the second step.

It shall be understood by persons skilled in the art that the two-step commitment scheme is not limited to the scheme described herein which involves, in the case of FIG. 7, sending a hash of the encrypted pre-master secret in step one, then the encrypted entire pre-master secret in step two. For example, it is also possible to break the first message into two parts, bitwise, send the first half encrypted under the server's public key in a first step, and then send the whole part later. Or indeed, any other ways of transmitting only partial information of a message which is encrypted under the server's public key in a first step, followed by the whole message in the second step are sufficient to achieve the aims of the present invention. The only requirements are that the first, partial message cannot be used to learn the whole message (a hash function, or any one-way function, can achieve this), the partial message is encrypted under the server's public key, and the partial message can be used to unambiguously identify the whole message (again, any one-way function can achieve this).

From an implementation point of view, in connection with the embodiment of FIG. 7, the present invention contemplates at least three methods of achieving such a "two-step delayed commitment" scheme.

(1) send the hash of an encrypted pre-master secret, interject a server nonce, followed by the encrypted pre-master secret;

(2) send half of the pre-master secret (bitwise) which is encrypted under the server's public key, interject a server nonce, followed by the encrypted pre-master secret; and

(3) send a predetermined portion of the pre-master secret which is encrypted under the server's public key, interject a server nonce, followed by the encrypted pre-

master secret.

In the embodiments of FIGS. 5 and 6, since it is the server nonce which is sent in two parts, the present invention likewise contemplates analogous methods for achieving the "two-step delayed commitment" scheme.

(1) send the hash of an encrypted server nonce, interceded by the server's certificate and the encrypted pre-master secret, followed by the encrypted server nonce;

(2) send half of the server nonce (bitwise) which is encrypted under the client's public key, interceded by the server's certificate and the encrypted pre-master secret, followed by the encrypted server nonce; and

(3) send a predetermined portion of the server nonce which is encrypted under the client's public key, interceded by the server's certificate and the encrypted pre-master secret, followed by the encrypted server nonce.

As described herein, two basic approaches for preventing an "ex-employee" type of attack on the conventional SSL/TLS protocol are presented which require only slight modification to the current protocol. Extra security is achieved by requiring the intruder to use an active and sustained man-in-the-middle attack for all subsequent (i.e. post-handshake) SSL/TLS communications, rather than being able to engage in a passive eavesdrop.

Although the invention has been described in the context of the most widely implemented "named-server anonymous-client" version of the SSL/TLS protocol, it will be readily appreciated by those skilled in the art that the same improvements and advantages can be added to a "named-server named-client" version of SSL/TLS to achieve similar results. Since the present invention requires little change to the original protocol, the invention can be made to co-exist with the current version of SSL/TLS and, as described above, can be used as an option which is added to the "negotiated cipher" specification of the current SSL/TLS protocol, wherein the improved version of the handshake protocol according to the present invention will be used only if both client and server agree to do so.

The present invention is also applicable to a more recent version of the handshake protocol, adopted in the wireless transport layer security (WTLS) which is an adaptation of TLS for wireless applications. Apart from all versions used in conventional TLS, the WTLS has introduced a further version of the handshake protocol called the "optimized full handshake" (OFH) protocol. In the OFH protocol, instead of receiving a certificate each time a new session of WTLS is initiated, for efficiency, the client can choose to reuse a certificate possessed in the client's local environment. This protocol, however, is obviously more at risk because freshness of certificates is sacrificed for efficiency. The scheme according to the present invention solves the problem of freshness, making the OFH protocol a more viable option.

Yet another advantage of the enhanced handshake protocol according to the present invention is that, since the client is not required to check the freshness of the server certificates, it brings both technical and legal benefits to clients who otherwise would bear the burden and responsibility for checking server certificate freshness. Thus, since there is no liability for failure to check certificate freshness, products and systems incorporating the enhanced protocol will achieve an important commercial benefit.

In terms of the environments in which the improved protocol shall be implemented and used, though not limited thereto, a typical environment would be the same as that discussed in Elgamal et al., U.S. Patent No. 5,657,390, wherein the handshake protocol is implemented as a sockets application program interface between the application and transport layers in a known IP sockets connection, such as the widely-used winsock transport control protocol (tcp). The implementation details for the improved protocol will not differ substantially from those of conventional SSL/TLS already disclosed by Elgamal et al., the disclosure of which has been incorporated herein by reference, and hence detailed discussion of such implementation details have been omitted from the present disclosure.

It shall be understood that various modifications will be apparent and can be easily made by persons skilled in the art without departing from the scope and spirit of the present invention. Accordingly, the following claims shall not be limited by the description or illustrations set forth herein, but shall be construed to cover with reasonable breadth all features which may be envisaged as equivalents by those skilled in the art.

What is claimed is:

1. A method of conducting secure communications using both symmetric and public key cryptography between two entities, typically a client and a server, comprising the steps of:

transmitting a message from a first entity to a second entity, wherein said message comprises partial information concerning a randomly generated pre-master secret which is used for generating a symmetrical master secret under which the secure communications are to be encrypted, said partial information including therein an encryption under a public key of said second entity;

transmitting at least one intervening message from said second entity to said first entity, said intervening message comprising a random component generated by the second entity; and

transmitting a further message from said first entity to said second entity, after receiving said intervening message from said second entity, said further message comprising the complete pre-master secret encrypted under said public key of said second entity,

wherein said pre-master secret cannot be learned from said partial information but said partial information can unambiguously identify said pre-master secret, and wherein any alteration to said complete pre-master secret encrypted under said public key of said second entity can be revealed.

2. The method according to claim 1, wherein said partial information comprises a hash of said pre-master secret encrypted under the public key of said second entity.

3. The method according to claim 1, wherein said partial information comprises a bitwise half of said pre-master secret encrypted under the public key of said second entity.

4. The method according to claim 1, wherein said partial information comprises a predetermined bitwise portion of said pre-master secret encrypted under the public key of said second entity.

5. A method of conducting secure communications using both symmetric and public key cryptography between two entities, typically a client and a server, comprising the steps of:

transmitting a message from a first entity to a second entity, wherein said message comprises the public key of a private-public key pair generated by said first entity;

transmitting a message from said second entity to said first entity, said message comprising partial information concerning a nonce of the second entity encrypted under the public key of said first entity;

transmitting a further message from said first entity to said second entity, said further message comprising a complete pre-master secret encrypted under a public key of said second entity; and

transmitting a further message from said second entity to said first entity, said further message comprising complete information concerning said nonce of the second entity encrypted under said public key of said first entity,

wherein said nonce of the second entity cannot be learned from said partial information but said partial information can unambiguously identify said nonce of the second entity, and wherein any alteration to said nonce of the second entity encrypted under said public key of said first entity can be revealed.

6. The method according to claim 5, wherein said partial information is a hash function of said nonce of said second entity encrypted under the public key of said first entity.

7. The method according to claim 5, wherein said partial information comprises a bitwise half of said nonce of said second entity encrypted under the public key of said first entity.

8. The method according to claim 5, wherein said partial information comprises a predetermined bitwise portion of said nonce of said second entity encrypted under the public key of said first entity.

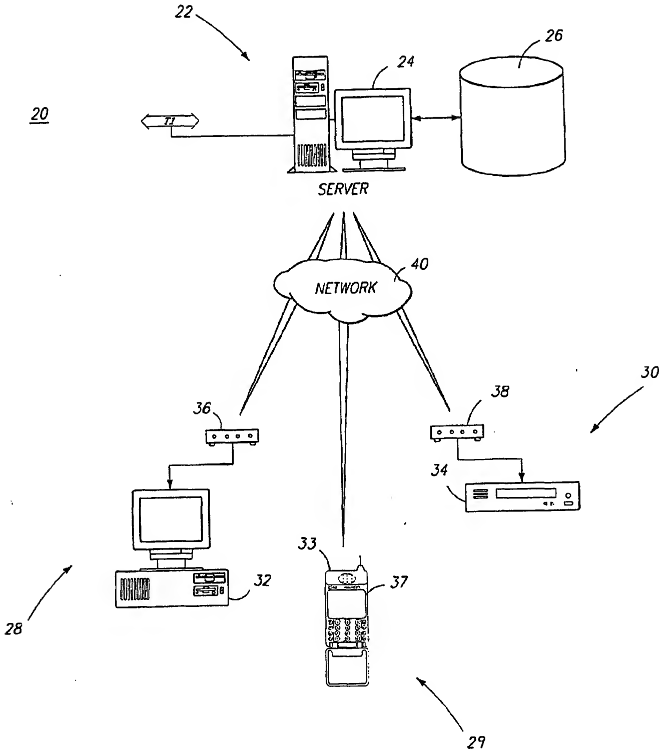


Fig 1

Direction	Message Content	Message No.
$C \rightarrow S$	(N_C, T_C)	M_1
$S \rightarrow C$	(N_S, T_S)	M_2
$S \rightarrow C$	$\{S, K_S^+\}_{K_{CA}^-}$	M_3
$C \rightarrow S$	$\{N_C^*\}_{K_S^+}$	M_4
$S \rightarrow C$	$\{H(K_{CS}, CS_5, (M_1, M_2; M_3, M_4))\}_{K_{CS}}$	M_5
$C \rightarrow S$	$\{H(K_{CS}, CS_6, (M_1, M_2; M_3, M_4))\}_{K_{CS}}$	M_6

FIG. 2

(Prior Art)

Direction	Message Content	Message No.
$C \rightarrow S$	(N_C, T_C)	M_1
$S \rightarrow C$	(N_S, T_S)	M_2
$S \rightarrow (C)I$	$\{S, K_S^+\}_{K_{CA}^-}$	M_3'
$(S)I \rightarrow C$	$\{S, K_S'^+\}_{K_{CA}^-}$	M_3''
$C \rightarrow (S)I$	$\{N_C^*\}_{K_S'^+}$	M_4'
$(C)I \rightarrow S$	$\{N_C^*\}_{K_S^+}$	M_4''
$S \rightarrow (C)I$	$\{H(K_{CS}, CS_5, (M_1, M_2, M_3', M_4''))\}_{K_{CS}}$	M_5'
$(S)I \rightarrow C$	$\{H(K_{CS}, CS_5, (M_1, M_2, M_3'', M_4'))\}_{K_{CS}}$	M_5''
$C \rightarrow (S)I$	$\{H(K_{CS}, CS_6, (M_1, M_2, M_3'', M_4'))\}_{K_{CS}}$	M_6'
$(C)I \rightarrow S$	$\{H(K_{CS}, CS_6, (M_1, M_2, M_3', M_4''))\}_{K_{CS}}$	M_6''

FIG. 3

Direction	Message Content	Message No.
$C \rightarrow S$	(N_C, T_C, K_C^+)	M_1
$S \rightarrow C$	$(\{N_S\}_{K_C^+}, T_S)$	M_2
$S \rightarrow C$	$\{S, K_S^+\}_{K_{CA}^-}$	M_3
$C \rightarrow S$	$\{N_C^*\}_{K_S^+}$	M_4
$S \rightarrow C$	$\{H(K_{CS}, CS_5, (M_1, M_2: M_3, M_4))\}_{K_{CS}}$	M_5
$C \rightarrow S$	$\{H(K_{CS}, CS_6, (M_1, M_2: M_3, M_4))\}_{K_{CS}}$	M_6

FIG. 4

Direction	Message Content	Message No.
$C \rightarrow S$	(N_C, T_C, K_C^+)	M_1
$S \rightarrow C$	$(H(\{N_S\}_{K_C^+}), T_S)$	M_2
$S \rightarrow C$	$\{S, K_S^+\}_{K_{CA}^-}$	M_3
$C \rightarrow S$	$\{N_C^*\}_{K_S^+}$	M_4
$S \rightarrow C$	$\{N_S\}_{K_C^+}$	$M_{4.1}$
$S \rightarrow C$	$\{H(K_{CS}, CS_5, (M_1, M_2, M_3, M_4, M_{4.1}))\}_{K_{CS}}$	M_5
$C \rightarrow S$	$\{H(K_{CS}, CS_6, (M_1, M_2, M_3, M_4, M_{4.1}))\}_{K_{CS}}$	M_6

FIG. 5

Direction	Message Content	Message No.
$C \rightarrow S$	(N_C, T_C)	M_1
$S \rightarrow C$	(N_S, T_S)	M_2
$S \rightarrow C$	$\{S, K_S^+\}_{K_{CA}^-}$	M_3
$C \rightarrow S$	$\{N_C^*\}_{K_S^+}$	M_4
$S \rightarrow C$	$\{N_S\}_{K_C^+}$	$M_{4.1}$
$S \rightarrow C$	$\{H(K_{CS}{:}\ CS_5, (M_1, M_2, M_3, M_4, M_{4.1}))\}_{K_{CS}}$	M_5
$C \rightarrow S$	$\{H(K_{CS}{:}\ CS_6, (M_1, M_2, M_3, M_4, M_{4.1}))\}_{K_{CS}}$	M_6

FIG. 6

Direction	Message Content	Message No.
$C \rightarrow S$	(N_C, T_C)	M_1
$S \rightarrow C$	(N_S, T_S)	M_2
$S \rightarrow C$	$\{S, K_S^+\}_{K_{CA}^-}$	M_3
$C \rightarrow S$	$H(\{N_C^*\}_{K_S^+})$	$M_{4.0}$
$S \rightarrow C$	N_S^*	$M'_{4.1}$
$C \rightarrow S$	$\{N_C^*\}_{K_S^+}$	M_4
$S \rightarrow C$	$\{H(K_{CS}, CS_5, (M_1, M_2, M_3, M_{4.0}, M_{4.1}, M_4))\}_{K_{cs}}$	M_5
$C \rightarrow S$	$\{H(K_{CS}, CS_6, (M_1, M_2, M_3, M_{4.0}, M_{4.1}, M_4))\}_{K_{cs}}$	M_6

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/08654

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04L 9/00
US CL : 713/171; 380/285
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 713/171; 380/285

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

West, Crypto Proceeding, EIC search

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5734720 A (SALGANICOFF) 31 March 1998, all	1-8
Y	US 5953424 A (VOGELESAND et al) 14 September 1999, all	1-8
Y	US 5412723 A (CANETTI et al) 02 May 1995, all	1-8

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents.

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 20 JUNE 2001	Date of mailing of the international search report 26 JUL 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer GAIL HAYES Peggy Harrod
Facsimile No. (703) 305-3230	Telephone No. (703) 308-4562